MISSION OPERATIONS AND DATA SYSTEMS DIRECTORATE

Renaissance Team Systems Engineering Facility (SEF) Activity Plan

February 1995

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

Renaissance Team
Systems Engineering Facility (SEF)
Activity Plan

February 1995

Prepared Under Contract NAS5-31500
Task Assignment 05-601

Prepared by:

Michael Bracken, Automation Technology Section Date
GSFC, Code 522.3

P. Joseph Gibson, Computer Sciences Corporation Date
GSFC, Code 504

Goddard Space Flight Center
Greenbelt, Maryland

Preface

Questions concerning this document should be addressed to:

Renaissance Project Manager
Code 504
Goddard Space Flight Center
Greenbelt, Maryland 20771

Contents

Preface

Section 1. Introduction

Section 2. SEF Overview

Section 3. Renaissance Testbed Activities

Contents (continued)

Appendix A. Planned SEF Prototyping Activities

Appendix B. Detailed Plan for SEF Activities

Acronyms and Abbreviations

Figures

Tables

Section 1. Introduction

This document presents an initial plan for the Mission Operations and Data Systems Directorate (MO&DSD) Systems Engineering Facility (SEF). The REusable Network Architecture for Interoperable Space Science Analysis, Navigation, and Control Environments (Renaissance) team has taken the lead in developing the SEF concept and managing the initial phases of SEF development.

1.1 Purpose

The purpose of this document is to present an overview of the SEF concept, present the layout of the initial SEF facility (often referred to as the Renaissance testbed), and describe the initial prototyping activities that will be accomplished in the SEF.

1.2 Scope

This document presents a brief overview of the SEF concept. Detailed SEF concept development is currently underway and will be made available in other documents. At least two documents, a project plan and an infrastructure plan, are currently under development. Additionally, this document presents details about the current SEF facility. The activities considered for the SEF are described, and the detail plan for the selected activities for fiscal year (FY) 1995 are provided.

1.3 Background

Renaissance is a systems engineering activity undertaken by MO&DSD to reengineer the development and maintenance of ground data systems (GDSs). The goal of Renaissance is to greatly reduce the development time and costs for deploying and maintaining a GDS. The goal for each new mission is to attain a high level of reuse of ground system building blocks (BBs)—well developed, functionally bounded ground system elements that adhere to agreed-upon standards.

During the early stages of Renaissance, it became apparent that analysis activities alone would not be sufficient to validate all Renaissance concepts. The notion of a "testbed" was introduced as a mechanism for prototyping and validating Renaissance concepts. The MO&DSD SEF grew out of this notion.

1.4 Document Organization

Section 2 of this document presents a concept overview and configuration of the SEF. Section 3 describes how the activities are chosen and the information detail provided. The appendices list the recommended activities, those selected for the initial SEF work, and the schedule for their implementation.

Section 2. SEF Overview

2.1 SEF Concept

The SEF concept is based on the idea of a centralized, dedicated facility for providing the following:

- Management of the MO&DSD BBs and ground system architectures
- Validation and integration of proposed BBs
- Rapid development of new mission architectures
- Support of new missions during early mission planning

To provide these and other services, the SEF will be segmented into the following four integrated environments:

- Repository
- Building Block Validation and Integration (BBVI)
- Architecture Engineering and Testbed (AET)
- Mission-Specific Support (MSS)

The SEF is to be partitioned into unique functional areas or environments. Each environment will have tools and resources for accomplishing particular tasks. If a given task requires resources from another environment, those resources will be allocated. The main goal of the SEF is to provide resources for any task that will help MO&DSD perform its function better, faster, and less expensively. An initial set of tasks is outlined below in the descriptions of each environment.

2.1.1 Repository

The repository will provide the resources for managing the MO&DSD BBs and ground system architectures. The key function of the repository is for configuration management of baseline BBs, the Renaissance architecture (current baseline), and prior mission architectures. Additionally, storage and management of other products like requirements traceability to BBs and metrics will be provided. BB maintenance will not be provided, as this is a function of the centers of expertise (COEs).

2.1.2 Building Block Validation and Integration

BBVI will provide resources for validating proposed BBs [including proposed commercial off-the-shelf (COTS) BBs] and establishing conformance to agreed-upon standards. The activities included will be at the BB level. Once a BB is validated, it will then be integrated into the baseline Renaissance architecture. Other BBVI activities are to identify BB limitations, evaluate COTS tools and products, evaluate and validate new tools, and identify, evaluate, and propose new standards.

2.1.3 Architecture Engineering and Testbed

The AET function will provide tools and resources for the rapid development of new mission architectures. The activities to be undertaken are end-to-end, architecture level and will include benchmarking and general evaluation of the effectiveness of an architecture. BBs can be pulled out and plugged into an architecture to determine the relative effectiveness of a given prototype architecture. Demonstrations of existing or new architectures to outside organizations will be provided within AET also.

2.1.4 Mission-Specific Support

The interface for new MO&DSD customers (the flight projects) is MSS. This area will support new missions during early mission planning. MSS support personnel will assist mission teams in evaluating new mission requirements to establish which BBs can be used to support the mission. Included in the analysis will be specific information about the availability, performance, and costs of BBs. MSS will rely heavily on the resources of all other environments in order to accomplish mission-related tasks.

The SEF is a cooperative facility. Whereas the Renaissance project will provide resources for managing the SEF and accomplishing various systems engineering tasks, the participation of all MO&DSD divisions is essential.

2.2 SEF Development Approach

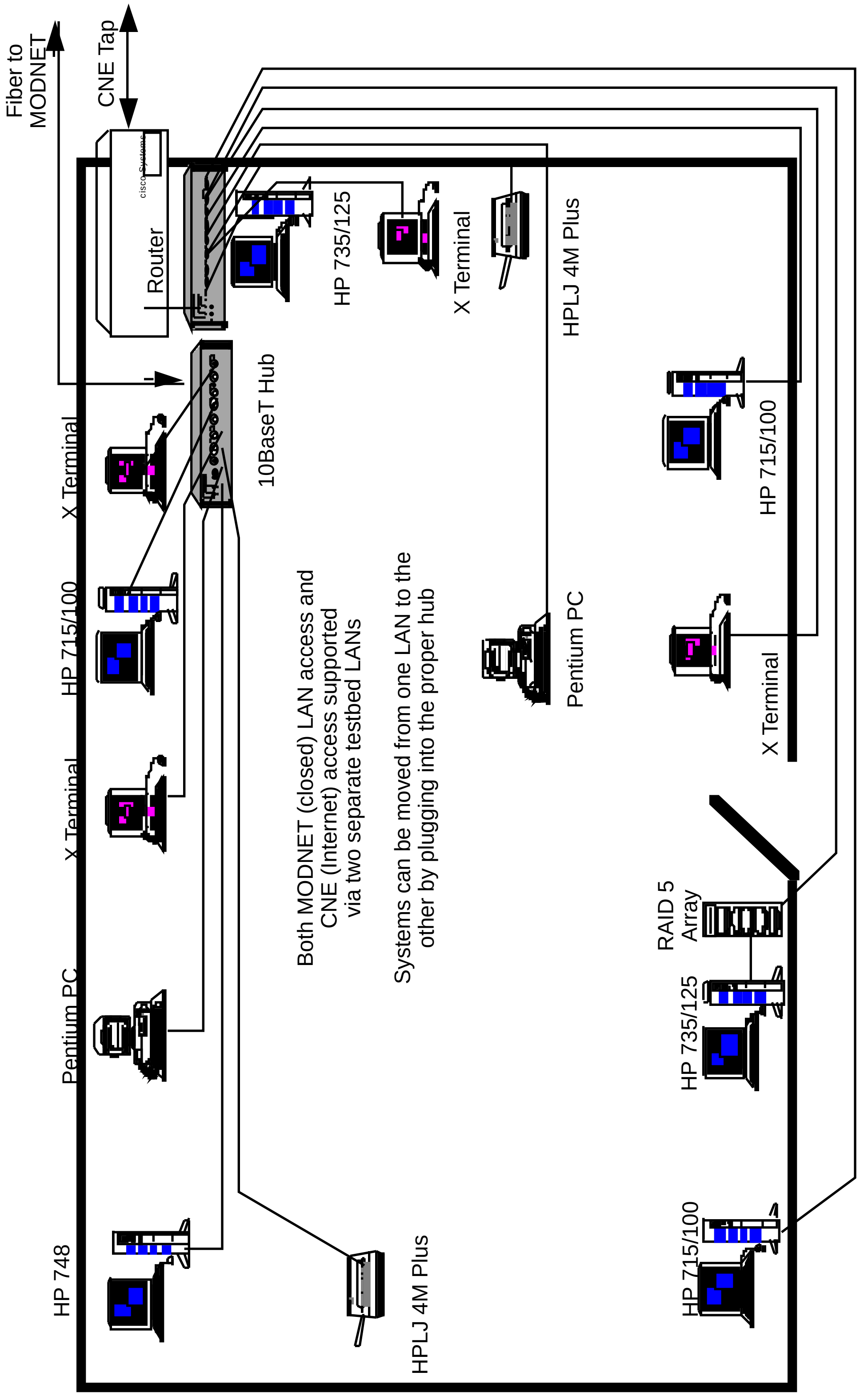
The current plan for SEF development is to employ a phased approach. Each phase will establish a new baseline with subsequent phases building on what exists. The first phase will establish an environment to accomplish several discrete prototyping activities and will evaluate standards. Planning for the second phase is underway.

Using the concept descriptions from above, phase one will establish the baseline BBVI environment. As of this writing, installation of equipment into the Goddard Space Flight Center (GSFC) Building 28 facility (Room N271) is about 75 percent complete. The Building 28 facility represents the first generation of the SEF. Once the facility is in place, several prototyping and development activities will commence. These activities are described in detail in the appendices.

2.3 Configuration

2.3.1 Hardware Configuration and Network Connectivity

The SEF room equipment configuration and connectivity are shown in Figure 2-1. It is expected that, as activities change, changes to this initial configuration will be necessary.



Both MODNET (closed) LAN access and CNE (Internet) access supported via two separate testbed LANs

Systems can be moved from one LAN to the other by plugging into the proper hub

Figure 2-1. SEF Architecture

2.3.2 Software Configuration

The following list constitutes the initial software for the SEF. Additional software will be procured as dictated by the specific activities undertaken in the testbed and will be identified with the activities.

- Hewlett-Packard Unix (HP-UX) operating system
- HP Programmers' Toolset
- HP FORTRAN 77 compiler
- HP C++ compiler bundled with Softbench
- Verdix Ada compiler for HP
- Oracle Version 7.0 for HP (8-user license)
- Builder Xcessory for HP (5-user license)
- Novell local area network (LAN) Workplace for DOS
- Age Logic XoftWare/32 for Windows
- HP C/American National Standards Institute (ANSI) compiler bundled with Softbench

2.3.3 Facility Layout

The SEF is located in GSFC Building 28, Room N271; access requires a GSFC keycard. Figure 2-2 shows the layout of the facility.

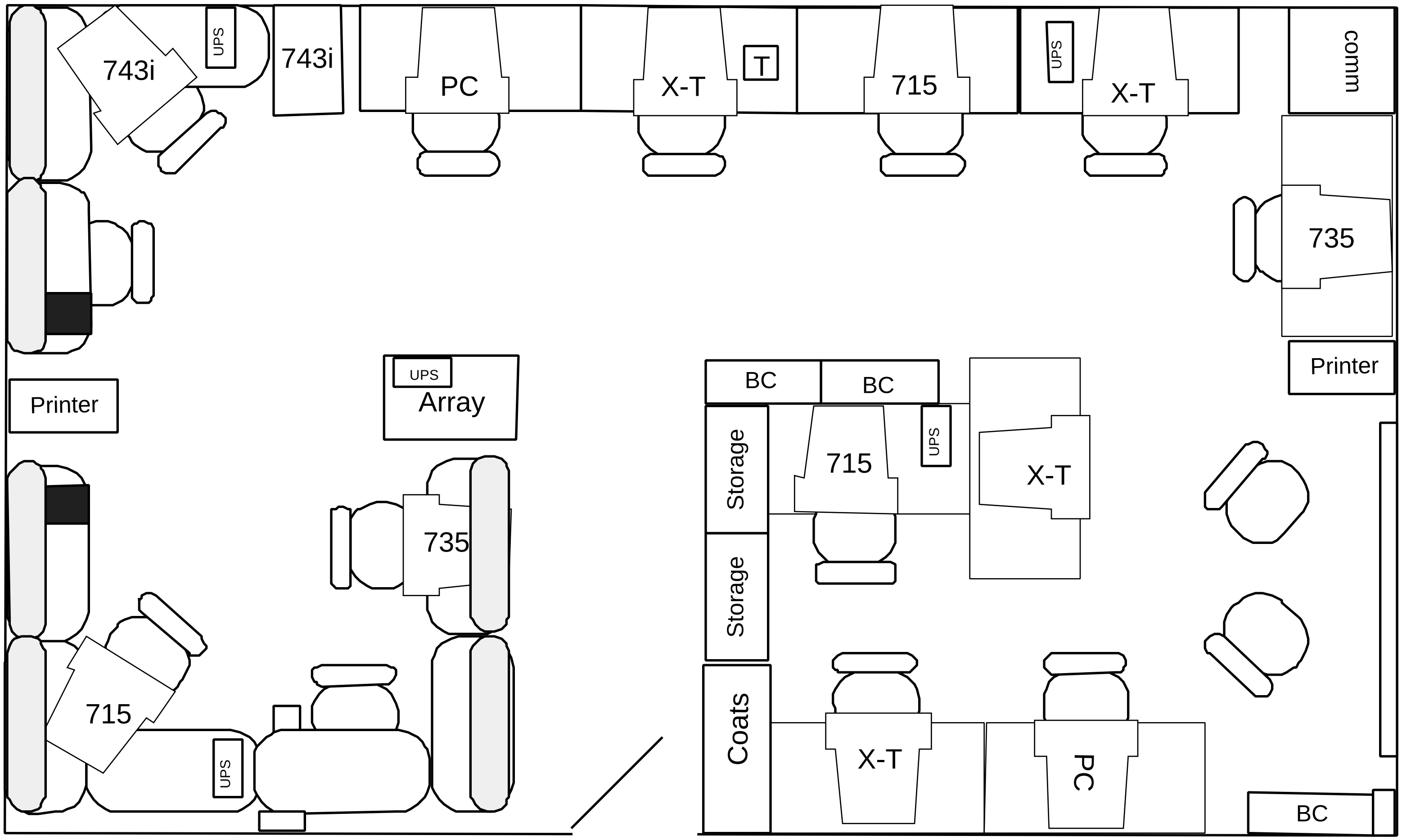


Figure 2-2. Testbed Facility Layout

Section 3. Renaissance Testbed Activities

3.1 Process for Proposing Renaissance Testbed Activities

SEF personnel surveyed Renaissance team members and compiled an initial list of activities to be considered for implementation in the SEF. The SEF team analyzed this list and has recommended an initial set of activities for implementation. Appendix A contains a list of both the larger list of activities considered for implementation and the smaller list of activities selected for implementation. Appendix B contains the detail plans for the selected activities.

On an ongoing basis, parties interested in and/or involved with the Renaissance effort are encouraged to propose new activities for implementation by SEF personnel. The SEF will act as a clearinghouse for these suggested activities; each proposed activity will be analyzed with respect to the goals of Renaissance and the effort required to implement the proposal.

The Renaissance Systems Engineering team is to provide overall direction for the Renaissance effort. The Systems Engineering team selects the standards and specifies the architecture of the Renaissance system. As a result, SEF activities will be selected and scheduled for implementation in consultation with the Renaissance Systems Engineering team and Renaissance management. A revised testbed activity plan that incorporates new approved activities will be published periodically.

3.2 Activity Plan Contents

Persons requesting action by the SEF will provide the following information for an activity:

- Purpose: Brief description of the purpose for the activity
- Originator: Renaissance team or organization requesting the activity
- Date: Date of request
- Description: Detailed description of the activity, including any background information necessary for understanding the request
- Existing Efforts: Description of current activities already underway that are related to the requested activity
- Priority: Originator's suggested priority (Renaissance management, in concert with the Systems Engineering team, will assign the priority for implementation of the activity.)
- Special Configuration Required: Any special hardware or software required beyond the nominal SEF configuration
- Contact(s): Sources of special information or knowledge of the proposed activity

After receiving a request, SEF personnel will analyze and provide the following information for each activity selected for implementation by the SEF team:

- Approach: The SEF team will describe in detail the technical approach to achieving the desired goals of the activity. The intent of this section is provide enough detail that the efficacy of the approach can be evaluated.
- Schedule: Dates when SEF activity will begin and end and, when appropriate, intermediate milestones for the activity will be provided.

3.3 Example of an Activity Description

This section provides an example of the information provided as part of an SEF activity. Appendix B provides the actual plans for the initial set of activities selected for implementation by the SEF.

Analyze Software Backplane Technologies

Purpose:

Explore the functionality and performance characteristics of three implementations of the Renaissance software backplane concept. The three implementations are

1. Distributed Computing Environment (DCE)
2. Structured Query Language (SQL)-database client/server middleware
3. Common Object Request Broker Architecture (CORBA)

The result of this activity is analysis of the performance, relative ease of implementation, and system administration requirements for each approach, as well as an analysis of how a combination of these technologies might implement the software backplane concept. The focus of this activity is on understanding the underlying technology rather than on the legacy software that may be ported to support this mission.

Originator: Systems Engineering

Date: 11/9/94

Description:

The Renaissance GDS architecture calls for use of a software backplane for connecting various system functions. The relative strengths and weaknesses of the three possible implementations of this concept must be explored.

Existing Efforts:

Code 510

The Transportable Payload Operations Control Center (TPOCC) generic software team and Mission Operations Center (MOC) technology integration task have performed some preliminary exploration and study of DCE.

Priority: High.

Special Configuration Required:

The SEF will need access to various implementations of SQL database middleware, DCE, and CORBA. Details of any special configuration required will be developed as part of the activity.

Contact(s): 510 (see existing efforts)

Approach:

Develop prototypes using each of the various approaches and the following guidelines:

- Study the vendor implementations for each backplane alternative. In consultation with the Systems Engineering team, select one or more vendor, and install the product in the SEF.
- Identify a combination of demonstration programs provided with the software and/or port parts of legacy software systems to create a prototype distributed system.
- Use Glance/Plus and other performance monitoring tools to benchmark the performance of the prototype. Measure and analyze the impact of system changes on system performance and resource utilization, especially with respect to changing the distribution of functions across platforms.
- Assess the relative difficulty of the administration of the product.

DCE

Explore the performance implications of dynamic and static binding of remote procedure calls (RPCs).

SQL-Database Middleware

Using legacy software [e.g., Mission Operations Planning Support System (MOPSS)] and/or a prototype application, develop one or more prototypes that incorporate multiple database management systems (DBMSs) and alternative DBMS products (i.e., not Oracle).

CORBA

TBS.

Schedule:

Milestone	Completion Date
DCE	
1. Install DCE evaluation copy	1/95
2. Run sample DCE program(s)	2/95
3. Issue DCE Report	4/95
SQL Database	
1. Issue plan for evaluating database backplane	5/95
2. Issue report on database backplane	9/95
CORBA	
1. Issue plan for evaluating CORBA	7/95

Appendix A. Planned SEF Prototyping Activities

A.1 SEF Activity Selection

Renaissance Team members identified a list of proposed prototyping activities for the SEF. The list was reviewed, updated at the Renaissance retreat in late September 1994, and further refined in subsequent working session between the SEF team and other Renaissance members. The resulting list of planned activities is summarized in Table A-1, including a cross-reference to the original, complete list of proposed prototyping activities.

A.2 Summary of Proposed Activities

Table A-2 summarizes all the proposed prototyping activities identified to date for the SEF and provides their relationship to the actual planned activities listed in Table A-1.

Table A-1. Summary of Planned SEF Prototyping Activities

Activity	Originator	Assigned Priority	Cross-Reference to Proposed Activities
1. Test front-end processor (FEP) elements on Unix platforms (e.g., spacecraft communications). Extend the prototype to connect with TPOCC and Packet Processor II (Pacor II) BBs.	Systems Engineering, Applications Engineering	High	1, 5, 6, 7, 8, 29, 22, 30
2. Analyze software backplane technologies, including DCE, CORBA, and SQL database.	Systems Engineering, Applications Engineering	High	4, 6, 9, 10, 17
3. Port TPOCC data server to DCE.	Applications Engineering	High	1, 4, 5, 8, 9, 10
4. Assess Renaissance BBs for X/Open compliance. Analyze the Specification 1170 test suite, and prototype an 1170-compliant subset of the Renaissance GDS function.	Systems Engineering	High	5, 10
5. Evaluate Renaissance approach to online help. Analyze and compare the Common Desktop Environment (CDE) and Mosaic help functions for possible use by Renaissance.	Applications Engineering	Medium	10
6. Explore integrated systems engineering tools. Investigate and evaluate tools that link online documents with a mission development database of requirements, review item dispositions (RIDs), configuration change requests (CCRs), etc. (Examples of such tools are KAPTUR, MSEE, PACE, and MDRSS.)	Process Engineering	Medium	10
7. Evaluate COTS products that implement BB functionality.	Systems Engineering	High	10
8. Examine system and network management tools and products.	SEF/Testbed	Medium	3, 10, 14, 15, 19, 25

Table A-2. Summary of Proposed SEF Prototyping Activities

No.	Activity	Suggested Priority	Prototype Plan Activity No.
1.	Integration of mission-specific applications using Renaissance BBs	H	1, 3
2.	Integration of ground system simulator (GSS) models into simulator	H	
3.	End-to-end fault isolation	H	8
4.	Selection of an industry standard software backplane	H	2, 3
5.	Porting of existing applications to new platforms/environments	H	1, 3, 4
6.	Ability of target platform to support GDS processing	H	1, 2
7.	Impact of Transmission Control Protocol (TCP)/Internet Protocol (IP) on Advanced Composition Explorer (ACE) system operations	H	1
8.	TPOCC performance FEP on target platform	H	1, 3
9.	Performance impacts of standard software backplanes	H	2, 3
10.	Evaluation of new products and technologies	H	2, 3, 4, 5, 6, 7, 8
11.	Integration of GSS into TPOCC environment	M	
12.	TPOCC graphical user interface (GUI) in simulator	M	
13.	Integration of TPOCC system test and operations language (TSTOL) with simulator	M	
14.	Operational setup of GDS from a central location	M	8
15.	Single-point GDS monitoring and control	M	8
16.	Deep Space Network (DSN) for fault isolation	M	
17.	Database design approach	M	2
18.	Evaluation of AlliedSignal Technical Services Corp. (ATSC's) "MATT" for use in Renaissance	M	
19.	End-to-end system monitoring	M	8
20.	Port TPOCC event server to simulator	M	
21.	Use of different hardware and software systems by GDS and simulator	M	
22.	End-to-end response times	M	1
23.	Support from remote sites	M	
24.	ACE GDS to Flight Dynamics Facility (FDF) interface	M	
25.	New advanced automation features	L	8
26.	Science Operations Center (SOC)/principal investigator (PI) emulation	H	
27.	Fanout	H	
28.	BB reuse library and evaluation metrics	M	
29.	Integration of event message handling by applications	M	
30.	Impact of network protocols	L	1

Appendix B. Detailed Plan for SEF Activities

The following sections describe in detail the activities currently selected for implementation in the Renaissance testbed.

B.1 Schedule of Activities

Figure B-1 shows the overall schedule for the planned implementation of SEF activities.

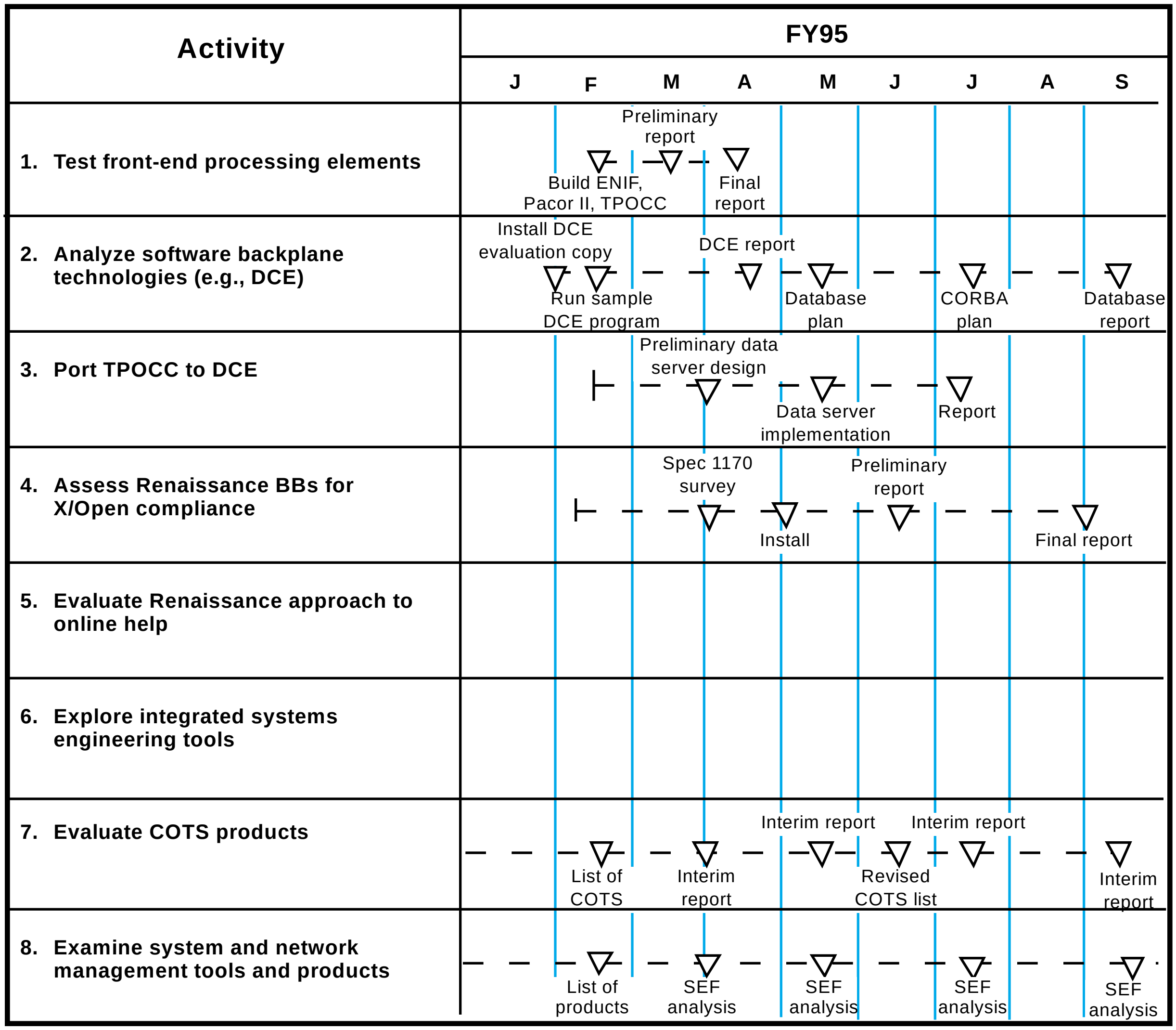


Figure B-1. Schedule of SEF Activities

B.2 SEF Activities

This section contains detailed plans for the SEF activities chosen for implementation in FY 1995.

B.2.1 Test Front-End Processing Elements

Purpose:

- Validate the ability of a standard Unix platform using standard communications protocols to perform FEP functions in a timely manner. Expand the prototype to include other functions of a GDS, and assess the performance of the resulting system.
- Refine the architecture and requirements of the spacecraft communications BBs.
- Assess the feasibility and effort associated with combining legacy systems.

Originator: Systems Engineering

Date: 11/17/94

Description:

Historically, telemetry data has been sent to a GDS using a special communications protocol [National Aeronautics and Space Administration (NASA) Communications (Nascom)] and was initially processed by special hardware called FEPs. Typically, FEPs extract packets from transfer frames. The Renaissance architecture calls for the use of standard communications protocols and workstations to perform some of these FEP functions. There is a need to validate the performance of this approach.

The ACE mission is the first mission to implement this portion of the Renaissance architecture. ACE will receive telemetry data from the DSN over a TCP/IP connection. Once received, this data will be passed to BBs from two legacy software suites: the TPOCC (for telemetry decommutation monitoring) and the Pacor II production data processor (PDP) subsystem (for science data processing). TPOCC and Pacor II both use TCP/IP socket connections for interprocess communication (IPC), but the application-level protocol of each system is quite different.

Existing Efforts:

Code 510

Code 510 is currently building a prototype front end for the ACE MOC. The architecture of this prototype is shown in Figure B-2. This prototype will execute across two HP 715 workstations connected by an Ethernet network.

One of the nodes houses the TPOCC Advanced Spacecraft Simulator (TASS), a DSN telemetry simulator that distributes virtual channel data units (VCDUs) using the standard DSN block (SDB) format. This simulator provides the VCDUs across a TCP/IP connection using the External Network Interface (ENIF).

The ENIF task is based on an existing TPOCC task, the TPOCC Nascom Interface (TNIF), which communicates to a set of custom hardware that handles the Nascom protocol. The standard libtnif functions that provide the interface between the TNIF and its clients uses TCP/IP. ENIF was created by modifying the TNIF software to interface with a TCP/IP socket to receive data, instead of using the custom Nascom hardware.

The ENIF provides the VCDUs to the Frame Logger and to the Packet Extractor tasks. The Packet Extractor extracts packets from the transfer frames and provides them to the Telemetry Decommutation task and other packet client tasks via a TCP/IP connection. Packet client tasks receive data from the packet extractor on a TCP/IP socket using a set of application programming interfaces (APIs) called libpkt.

The TPOCC prototyping task in Code 510 is scheduled to provide ENIF by January 15, 1995, for use to support an early software release to ACE integration and test (I&T) users.

Code 560

Code 560 personnel recently completed a port of the Pacor II software to the HP platform. The generic telemetry simulator (GTSIM) Transmit task simulates a Pacor II front end sending data across a TCP/IP network to the back-end processors. The production data processing subsystem (PDPS) is used for level-zero processing of Pacor II, and the real-time output subsystem (RTOS) is used for real-time routing of packets from Pacor II to outside users. GTSIM user simulator (UserSIM) simulates a real-time user receiving these packets across the TCP/IP network. Pacor II uses a set of APIs for setting up server and client sockets within PDPS and RTOS. For the purposes of the front-end prototype, only PDPS needs to be considered.

Priority: High

Special Configuration Required: None.

Contact(s): TPOCC, Pacor II

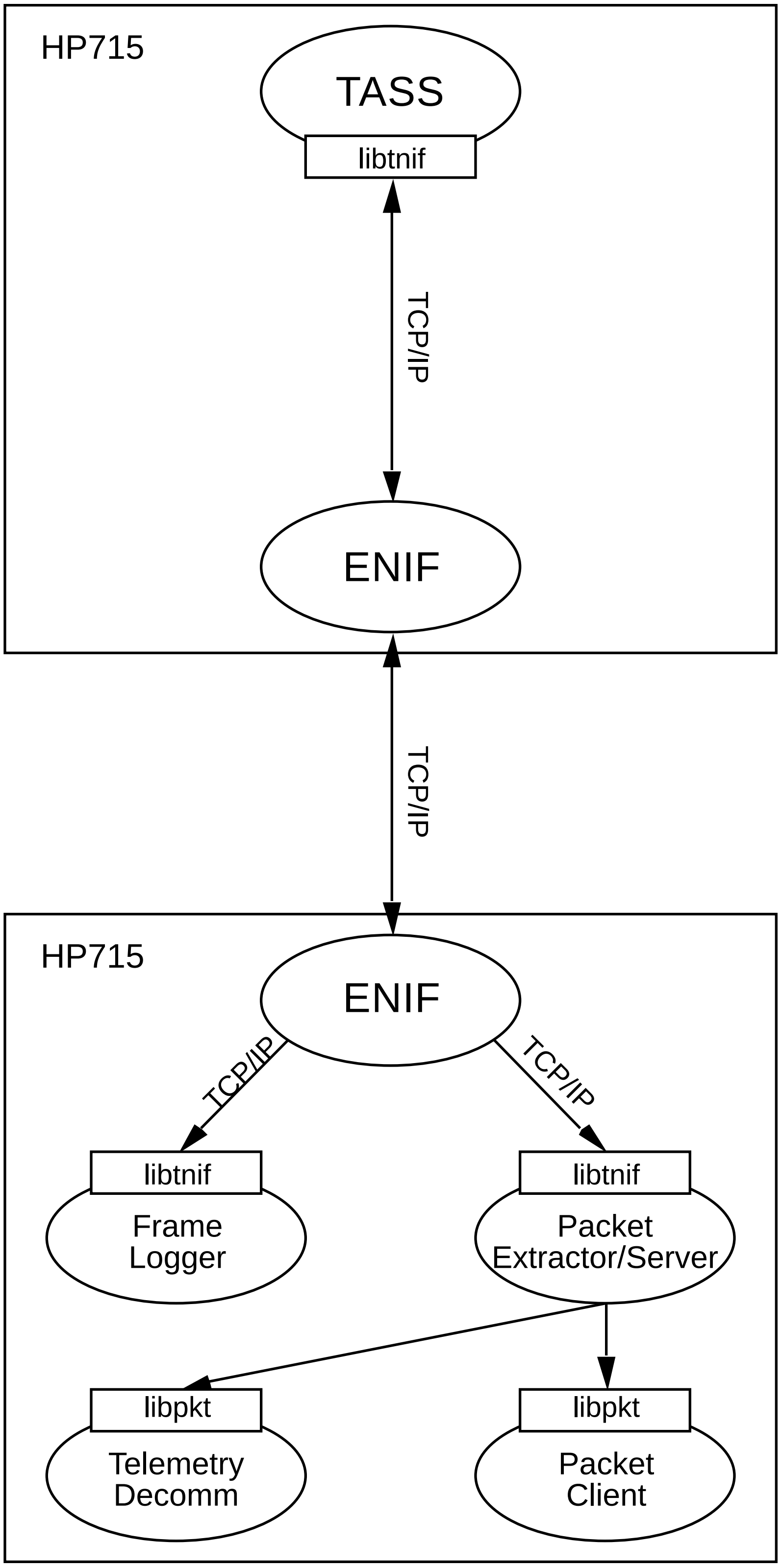


Figure B-2. ACE MOC Front-End Prototype

Approach:

1. Leverage off of the work being performed for ACE I&T. Use this prototype to benchmark performance, and verify the ability of standard products to process telemetry in TCP/IP packets.
2. Extend the prototype to include the TPOCC and Pacor II functions. Analyze the feasibility of a common API for these two legacy systems. Prototype and implement an approach to a common API and benchmark the resulting system.
3. Refine the requirements and architecture for the spacecraft communication BBs. Consider the inclusion of a new "DSN interface" BB before the packet server and frame synchronizer to shield these generic BBs from the DSN-specific characteristics of the new TCP/IP interface.
4. Begin development of specifications for the final software interface with spacecraft communications. Personnel from both the TPOCC and Pacor II projects should be involved to help analyze the impacts to legacy software. Existing APIs within the legacy systems should be considered as candidate interfaces.

Schedule:

Milestone	Completion Date
1. Build ENIF, Pacor II, and TPOCC	2/95
2. Issue Preliminary Performance Report	3/95
3. Issue Final Performance Report	4/95

B.2.2 Analyze Software Backplane Technologies

Purpose:

Explore the functionality and performance characteristics of three implementations of the Renaissance software backplane concept: DCE, SQL-database client/server middleware, and CORBA. The result of this activity is an analysis of the performance, relative ease of implementation, system administration requirements for each product, and how these technologies might work together to implement the backplane concept.

The focus of this activity is on understanding the underlying technology rather than on the legacy software that may be ported to support this activity.

Originator: Systems Engineering

Date: 11/9/94

Description:

The Renaissance GDS architecture calls for use of a software backplane for connecting various system functions. The relative strengths and weaknesses of the three possible implementations of this concept must be explored.

Existing Efforts:

Code 510

The TPOCC Generic Software team and MOC Technology Integration task have performed some preliminary exploration and study of DCE.

Code 504

The EOS Support task has studied DCE and CORBA as possible standards to be used on EOS.

Priority: High

Special Configuration Required:

The testbed will need access to various implementations of DCE, SQL database middleware, and CORBA.

Contact(s): None.

Approach:

Develop prototypes using each of the various approaches using the following guidelines:

- Study the vendor implementations for each backplane alternative. In consultation with the Systems Engineering team, select one or more vendor, and install the product in the SEF.
- Identify a combination of demonstration programs provided with the software and/or port parts of legacy software systems to create a prototype distributed system.

- Use Glance/Plus and other performance monitoring tools to benchmark the performance of the prototype. Measure and analyze the impact of system changes on system performance and resource utilization, especially with respect to changing the distribution of functions across platforms.
- Assess the relative difficulty of the administration of the product.
- Evaluate how the knowledge gained in the course of the investigation might affect the investigation of the other backplane technologies.

This general approach will be tailored for the specific backplane technology in the following ways:

<u>DCE</u>

Explore the performance implications of dynamic and static binding of RPCs.

<u>SQL-Database Middleware</u>

Using legacy software (e.g., MOPSS) and/or a prototype application, develop one or more prototypes that incorporate multiple DBMSs and alternative DBMS products (i.e., not Oracle).

<u>CORBA</u>

TBS.

<u>Schedule</u>:

Milestone	Completion Date
DCE	
1. Install DCE evaluation copy	1/95
2. Run sample DCE program(s)	2/95
3. Issue DCE Report	4/95
SQL Database	
1. Issue plan for evaluating database backplane	5/95
2. Issue report on database backplane	9/95
CORBA	
1. Issue Plan for evaluating CORBA backplane	7/95

B.2.3 Port TPOCC to DCE

Purpose:

Estimate the effort (time and staff) necessary to port TPOCC IPCs from network sockets to Open Software Foundation's (OSF's) DCE. This will put the TPOCC software on the "porting path" of an industry standard IPC model. An explicit goal of this activity is to develop an approach that allows TPOCC to migrate to DCE while minimizing disruption to TPOCC APIs.

Originator: Applications Engineering

Date: 11/9/94

Description:

The TPOCC software is implemented using the relatively low-level IPC model of network sockets. Although in the long run it is likely that TPOCC will be redesigned to incorporate a new IPC model, the Renaissance effort must develop a short- to medium-term design for missions that must be supported in the next 5 to 7 years that will rely on porting existing legacy systems to workstations. Steps must to taken to ensure that the IPC model used by TPOCC will remain supported during this timeframe. By replacing the network socket interface with DCE, the TPOCC software would be assured of support for its IPC model for the next 5 to 10 years (or more).

Existing Efforts:

Code 510

The TPOCC Generic Software task and X-ray Timing Explorer (XTE) MOC Insertion task are exploring DCE. Computer Sciences Corporation (CSC) expects to receive an evaluation copy of HP's implementation of DCE some time in early January 1995.

Priority: High

Special Configuration Required: DCE client/server software.

Contact(s): None.

Approach:

1. Write a new DCE-based data server for handling asynchronous data requests. (Asynchronous data requests are simple one-time requests for data, as opposed to a "subscription" for an ongoing stream of data.) Port parts of an existing TPOCC implementation or develop a driver to exercise this new DCE functionality. Run benchmark tests to compare the performance of the DCE implementation with the performance of the socket-based implementation.
2. Based on the knowledge gained from step 1, identify additional prototyping activities that are necessary to understanding the impact of DCE on TPOCC.

3. Develop a transition plan for how the TPOCC system might transition to DCE. (An explicit goal of this transition plan will be to minimize the impact of the migration to DCE on existing TPOCC APIs.) Include in the plan how much of the existing TPOCC code would change and how it would change. Estimate the time and staffing and other costs of implementing the transition plan.

Schedule:

Milestone	Completion Date
1. Review initial data server design	3/95
2. Complete data server implementation	5/95
3. Report and assessment of data server port	7/95

B.2.4 Assess Renaissance Building Blocks for X/Open Compliance

Purpose:

Determine the extent to which Renaissance BBs conform to the X/Open Unix standard, and analyze the performance implications of changes that may be necessary to comply with the standard.

Originator: Systems Engineering

Date: 11/17/94

Description:

It is important that the Renaissance BBs remain as portable as possible. Specification 1170 was developed by the X/Open committee to determine the extent to which X software complies with standard Unix system calls. There could be performance implications for modifications necessary for compliance with the X Open standard.

Existing Efforts: None.

Priority: High.

Special Configuration Required: None.

Contact(s): None.

Approach:

1. Survey the availability of tools that implement Specification 1170, and acquire the tools for the Renaissance testbed.
2. Identify a subset of a legacy system or Renaissance BB(s) for modification and benchmarking against Specification 1170. Prototype the subset and install it in the SEF.
3. Benchmark the performance of the system. Evaluate the performance implications of any modifications necessary to bring the system into compliance with Specification 1170.
4. Acquire platforms from other vendors and benchmark performance.

Schedule:

Milestone	Completion Date
1. Survey availability of Spec 1170	3/95
2. Install Spec 1170 implementation in SEF	4/95
3. Issue preliminary report	6/95
4. Issue final report	8/95

B.2.5 Evaluate Renaissance Approach to Online Help

Purpose:

Identify an approach for developing online help that can be applied consistently by Renaissance BB developers. The two goals that need to be simultaneously met with as little duplication of effort as possible are

1. Online help should be integrated with the CDE help system.
2. End-user documentation should be accessible to WorldWide Web (WWW) browsers.

Originator: Applications Engineering

Date: 1/17/94

Description:

The Renaissance user services working group has specified that all Renaissance user interface applications should migrate to CDE. CDE has a standard help system that allows each application's online help to be registered as a "help volume" that is accessible through a standard help viewer.

The CDE help system does not permit electronic access to the end-user information over Internet. This requirement is best met by installation of WWW servers publishing this information. Unfortunately, CDE help is written in one dialect of Standard Generalized Markup Language (SGML), namely HelpTag, while WWW uses Hypertext Markup Language (HTML). Thus, it is not clear how online help can be written that is visible through both CDE and Mosaic.

Existing Efforts:

The Renaissance user services working group performed some initial investigation into this problem. Names of several vendors building products to enhance the CDE desktop have been obtained.

Priority: Medium.

Special Configuration Required: None.

Contact(s): None.

Approach:

1. Investigate whether a commercial or public domain utility to convert from CDE HelpTag format to HTML (and vice versa) is or will be available in the near future. Start with vendor contacts.
2. Based on information from item #1, document an approach for creating online help for Renaissance BBs. Should the documentation first be written in HTML and then converted to HelpTag, or vice versa? List any tools that are available to aid this process.
3. Demonstrate a partial online help system based on applying the approach from #2.

Schedule: TBS.

B.2.6 Explore Integrated Systems Engineering Tools

Purpose:

Explore the feasibility of an integrated set of systems engineering tools that would link Renaissance documentation with a mission development database of requirements, RIDs, CCRs, etc.

Originator: Process Engineering

Date: 11/17/94

Description:

Traditional approaches for system development have not supported an integrated view of all mission development information. Each document has required manual updating when information or status of a related item has changed. For example, when a requirement paragraph numbering scheme has changed, developers have had to manually identify and update all related references in a project's documentation, which is labor-intensive and costly. Investigate and evaluate tools that link online documents with a mission development database of requirements, RIDs, CCRs, etc.

Existing Efforts:

Existing efforts will be identified as part of the activity.

Priority: Medium.

Special Configuration Required: None.

Contact(s): None.

Approach:

1. Survey existing efforts to develop systems of the type described above. (See Description of Existing Efforts.)
2. Develop a detailed approach and schedule for SEF action.

Schedule: To be developed as part of plan.

B.2.7 Evaluate COTS Products

Purpose:

Evaluate the functionality, product support, ease of use, and performance characteristics of COTS products that implement Renaissance BB functionality.

Originator: Systems Engineering

Date: 11/9/94

Description:

The Renaissance team has identified a set of reusable BBs as part of its GDS architecture. Renaissance also stresses the importance of providing maximum flexibility in the make/buy trades for implementing BBs. This activity will ensure that Renaissance acquires and maintains current knowledge of commercially available products that implement Renaissance BB functionality.

Existing Efforts:

Most divisions within Systems, Engineering, and Analysis Support (SEAS) and NASA/GSFC have teams that evaluate COTS products. An important part of the SEF COTS evaluation effort is to learn about and leverage existing efforts to evaluate COTS products.

Priority: High

Special Configuration Required:

Special configuration requirements will be largely product dependent. In general, the SEF will be evaluating products requiring hardware and software that is part of the nominal SEF configuration.

Contact(s): Contacts will be identified as part of this activity.

Approach:

1. Compile a list of COTS products that the various Renaissance teams have requested be evaluated. (A preliminary list of products has already been compiled.)

2. Survey existing COTS evaluation efforts around GSFC. To the extent possible, leverage knowledge gained in these evaluations to minimize duplication of effort.

3. Purchase or procure evaluation copies of the products in the list, and install them in the SEF. Develop prototypes and/or run sample programs provided as part of the product, and assess its applicability to Renaissance.

4. Where appropriate, use Glance/Plus and other performance monitoring tools to benchmark the performance of the product under evaluation.

5. Assess the relative difficulty of the administration of the product and the quality of support offered by the vendor.

Schedule:

Milestone	Scheduled Completion Date
List a set of COTS products to be evaluated.	2/95, 6/95
Issue an interim report of COTS products evaluated to date.	3/95, 5/95, 7/95, 9/95

B.2.8 Examine System and Network Management Tools and Products

Purpose:

Evaluate the SEF 's ability to support Renaissance (see Section 2). On an ongoing basis, propose changes that will improve service to SEF users.

Originator: SEF/Testbed

Date: 12/12/94

Description:

Standardized system and network management protocols and applications are planned to be used as tools to monitor and control the ground system computers, applications, and communications devices in Renaissance systems. The SEF will use COTS products both to evaluate and demonstrate the capabilities of such COTS products. HP Open View is one such product expected to be used. Because there are numerous third-party applications that work with Open View, ongoing assessment and evaluation is anticipated for this area of investigation. (That fact is reflected in the schedule, which shows frequent reassessments of where we are and where we are going.)

The correct combination of network utilities and system and network management products can automate functions that are currently manual, with a reduction in the personnel resources required. Using these products in the testbed will aid in its administration and configuration management.

Existing Efforts: None.

Priority: Routine.

Special Configuration Required: Not Applicable.

Contact(s): None.

Approach:

1. Identify additional products that will facilitate the maintenance and administration of the SEF. Procure and analyze the products. Periodically reassess the SEF configuration.

Schedule:

Milestone	Scheduled Completion Date
Identify initial products for use.	2/95
Assess SEF functionality, and identify new hardware and software to improve support of SEF users.	3/95, 5/95, 7/95, 9/95

Acronyms and Abbreviations

ACE	Advanced Composition Explorer
AET	Architecture Engineering and Testbed
ANSI	American National Standards Institute
API	application programming interface
ATSC	AlliedSignal Technical Services Corporation
BB	building block
BBVI	Building Block Validation and Integration
CCR	configuration change request
CDE	Common Desktop Environment
CNE	Center Networking Environment
COE	center of expertise
CORBA	Common Object Request Broker Architecture
COTS	commercial off-the-shelf
CSC	Computer Sciences Corporation
DBMS	database management system
DCE	Distributed Computing Environment
DOS	disk operating system
DSN	Deep Space Network
ENIF	External Network Interface
FDF	Flight Dynamics Facility
FEP	front-end processor
FORTRAN	formula translator language
FY	fiscal year
GDS	ground data system
GSFC	Goddard Space Flight Center
GSS	ground system simulator
GTSIM	generic telemetry simulator

GUI	graphical user interface
HP	Hewlett-Packard
HPLJ	Hewlett-Packard LaserJet
HTML	Hypertext Markup Language
I&T	integration and test
IP	Internet Protocol
IPC	interprocess communication
LAN	local area network
Mbps	megabits per second
MO&DSD	Mission Operations and Data Systems Directorate
MOC	Mission Operations Center
MODNET	MO&DSD Operational/Developmental Network
MOPSS	Mission Operations Planning Support System
MSS	Mission-Specific Support
NASA	National Aeronautics and Space Administration
Nascom	NASA Communications
OSF	Open Software Foundation
Pacor II	Packet Processor II
PC	personal computer
PDP	production data processor
PDPS	production data processing subsystem
PI	principal investigator
RAID	redundant arrays of independent disks
Renaissance	REusable Network Architecture for Interoperable Space Science Analysis, Navigation, and Control Environments
RID	review item disposition
RPC	remote procedure call
RTOS	real-time output subsystem
SDB	standard DSN block
SEAS	Systems, Engineering, and Analysis Support

SEF	Systems Engineering Facility
SGML	Standard Generalized Markup Language
SOC	Science Operations Center
SQL	Structured Query Language
TASS	TPOCC Advanced Spacecraft Simulator
TBS	to be scheduled
TCP	Transmission Control Protocol
TNIF	TPOCC Nascom Interface
TPOCC	Transportable Payload Operations Control Center
TSTOL	TPOCC system test and operations language
UPS	uninterruptible power supply
UserSIM	user simulator
UX	Unix
VCDU	virtual channel data unit
WAN	wide area network
WWW	Worldwide Web
XTE	X-ray Timing Explorer